Information-theoretic approaches

Uses the distribution of the minimum

$$p_{min}(x) \equiv p[x = \arg\min f(x)] = \int_{f:I\to\Re} p(f) \prod_{\substack{\tilde{x}\in I \\ \tilde{x}\neq x}} \theta[f(\tilde{x}) - f(x)]df$$

where θ is the Heaviside's step function. No closed form!

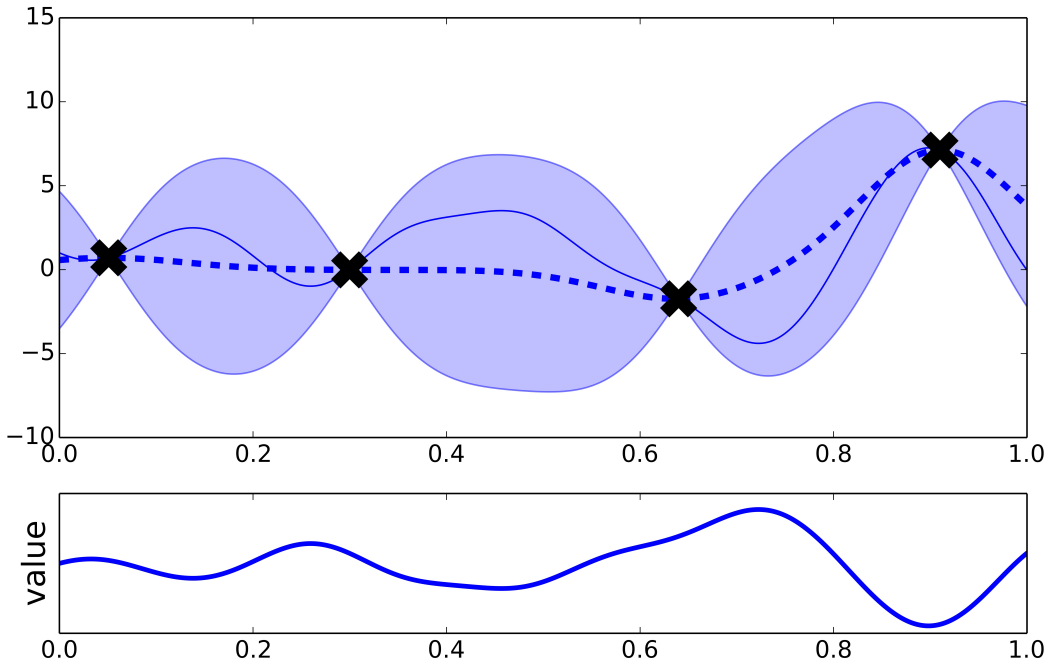
Use Thomson sampling to approximate the distribution. Generate many sample paths from the GP, optimize them to take samples from $p_{min}(x)$.

Thomson sampling

Probability matching

$$\alpha_{THOMSON}(\mathbf{x}; \theta, \mathcal{D}) = g(\mathbf{x})$$

$g(\mathbf{x})$ is sampled form $\mathcal{GP}(\mu(x), k(x, x'))$



Thompson sampling

Probability matching [Rahimi and B. Recht, 2007]

- It is easy to generate posterior samples of a GP at a finite set of locations.
- More difficult is to generate 'continuous' samples.

Possible using the Bochner's lemma: existence of the Fourier dual of k, $s(\omega)$ which is equal to the spectral density of k

$$k(x, x') = \nu \mathbb{E}_\omega \left[e^{-i\omega^T (x - x')} \right] = 2\nu \mathbb{E}_{\omega, b} \left[\cos(\omega x^T + b) \cos(\omega x^T + b) \right]$$

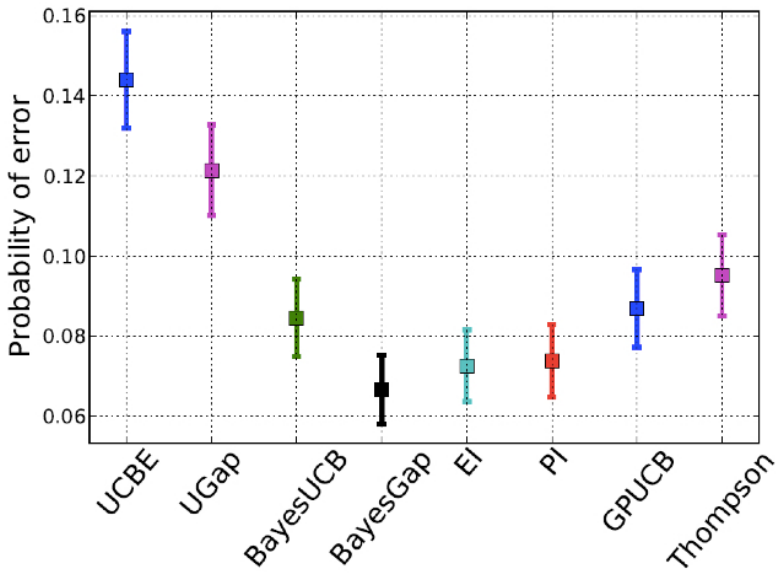
With sampling and this lemma (taking $p(w) = s(\omega)/\nu$ and $b \sim \mathcal{U}[0, 2\pi]$) we can construct a feature based approximation for sample paths of the GP.

$$k(x, x') \approx \frac{\nu}{m} \sum_{i=1}^{m} e^{-i\omega^{(i)T} x} e^{-i\omega^{(i)T} x'}$$

The choice of utility matters

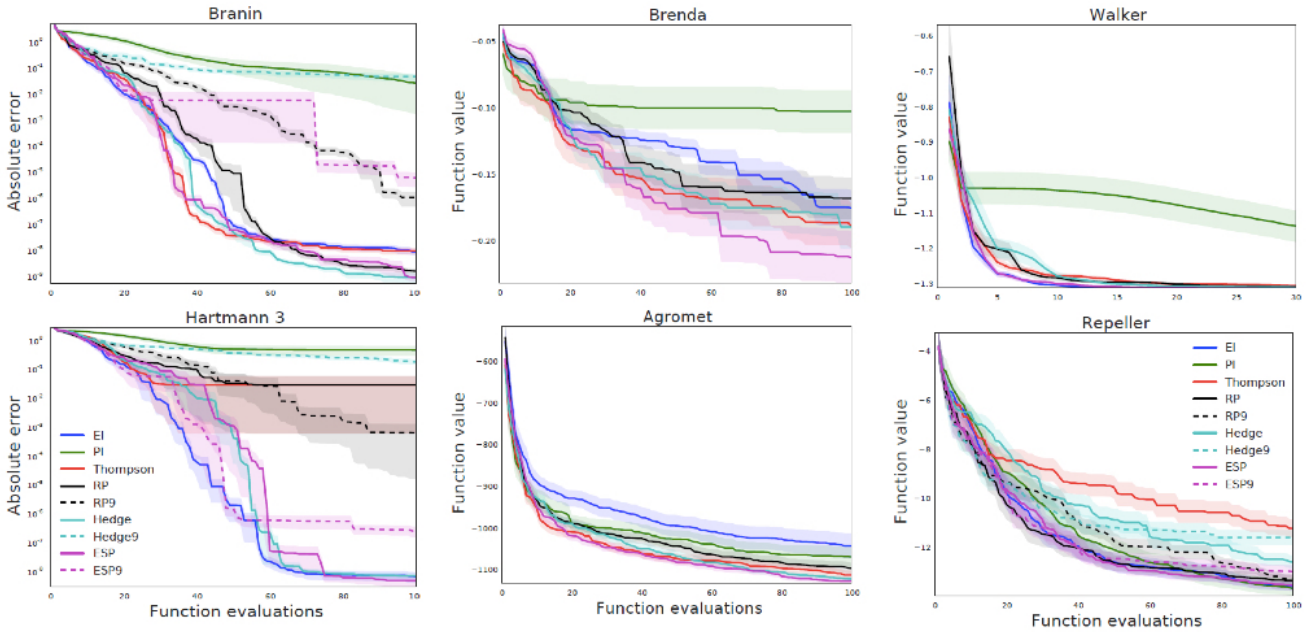
[Hoffman, Shahriari and de Freitas, 2013]

The choice of the utility may change a lot the result of the optimisation.



The choice of utility in practice

[Hoffman, Shahriari and de Freitas, 2013]



The best utility depends on the problem and the level of exploration/exploitation required.

Illustration of BO

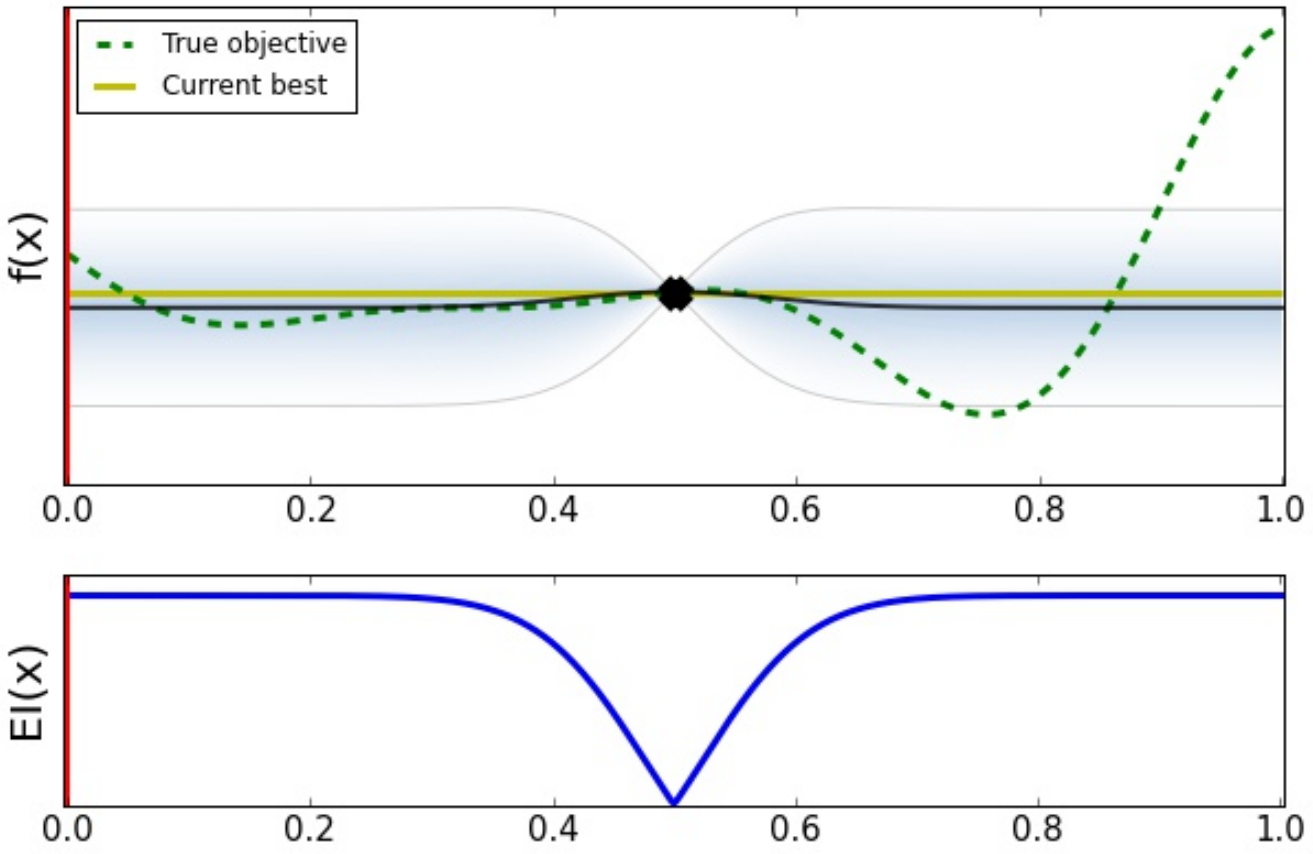


Illustration of BO

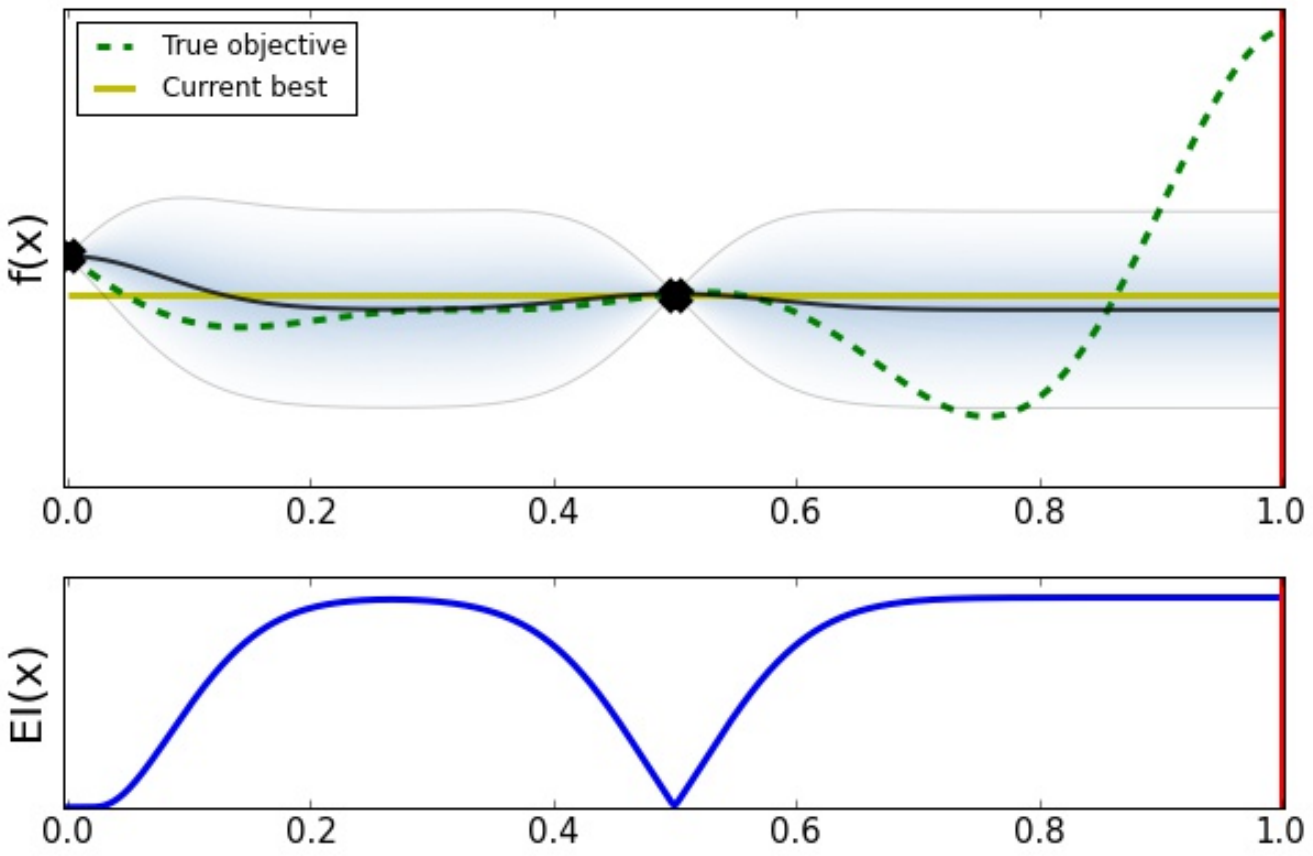


Illustration of BO

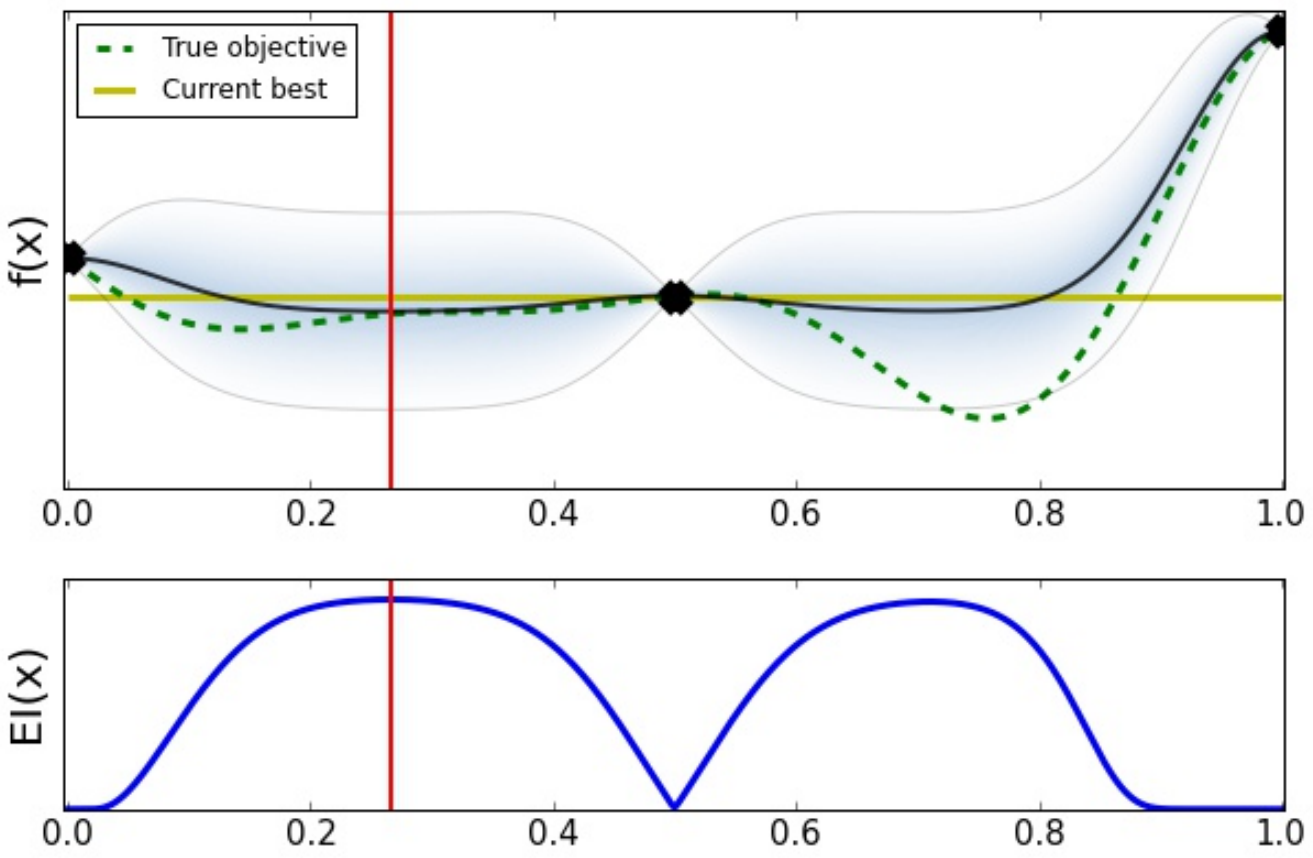


Illustration of BO

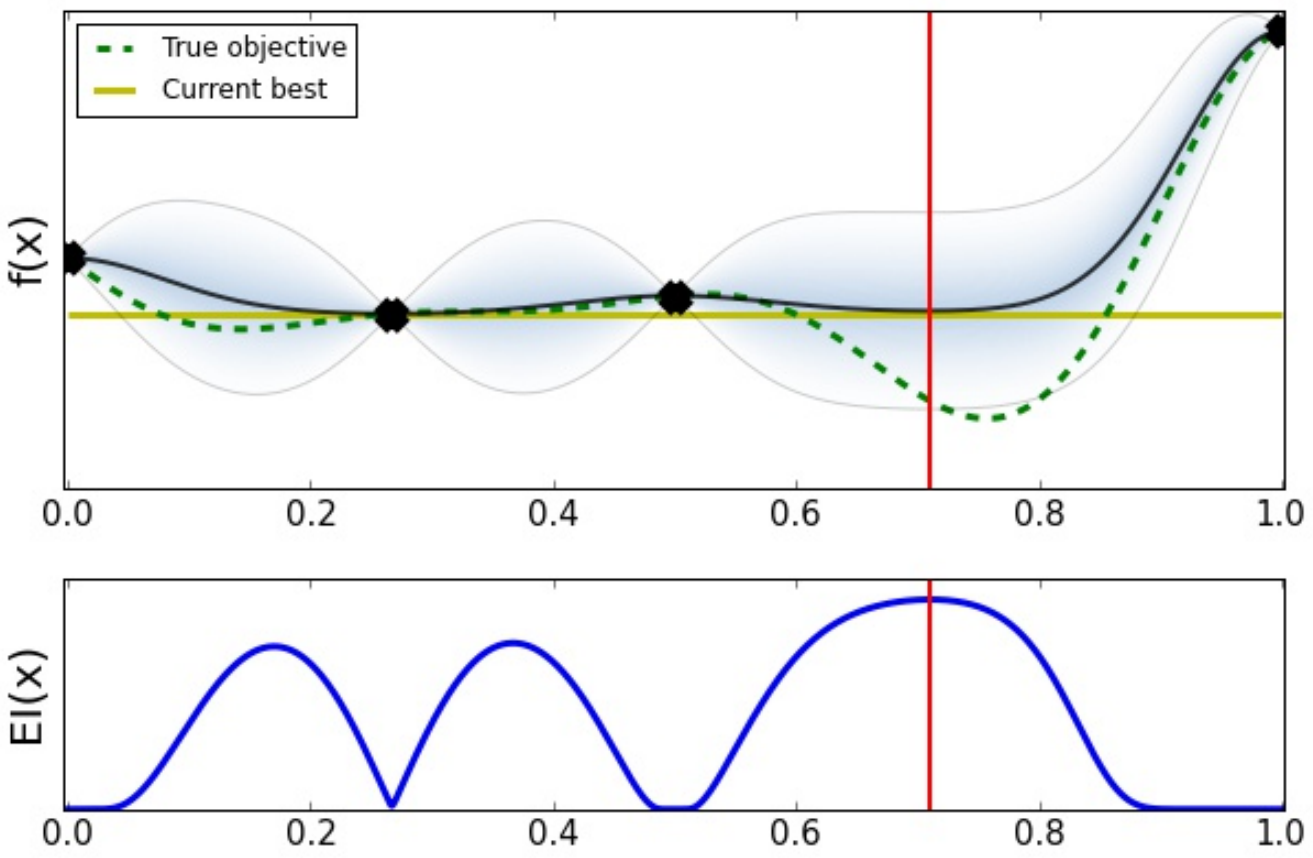


Illustration of BO

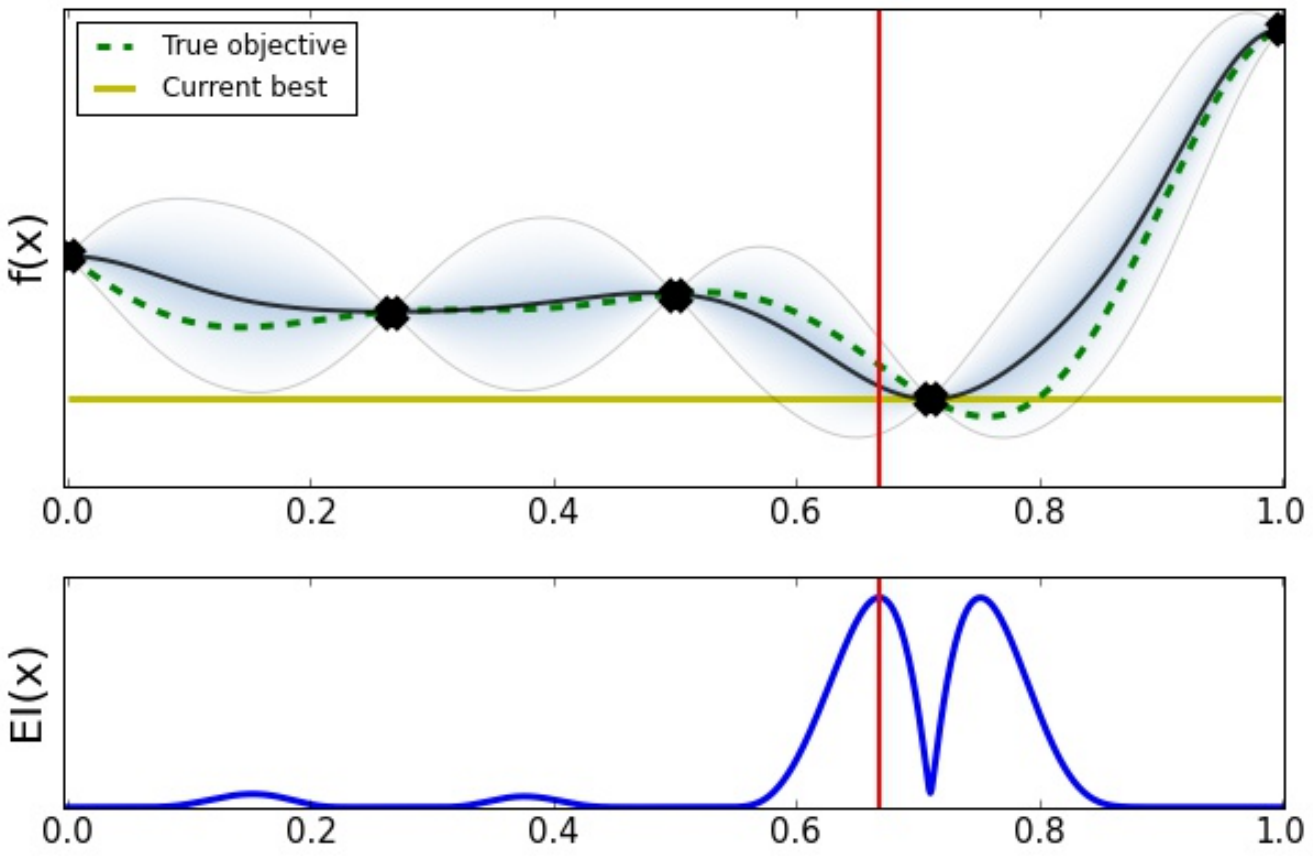


Illustration of BO

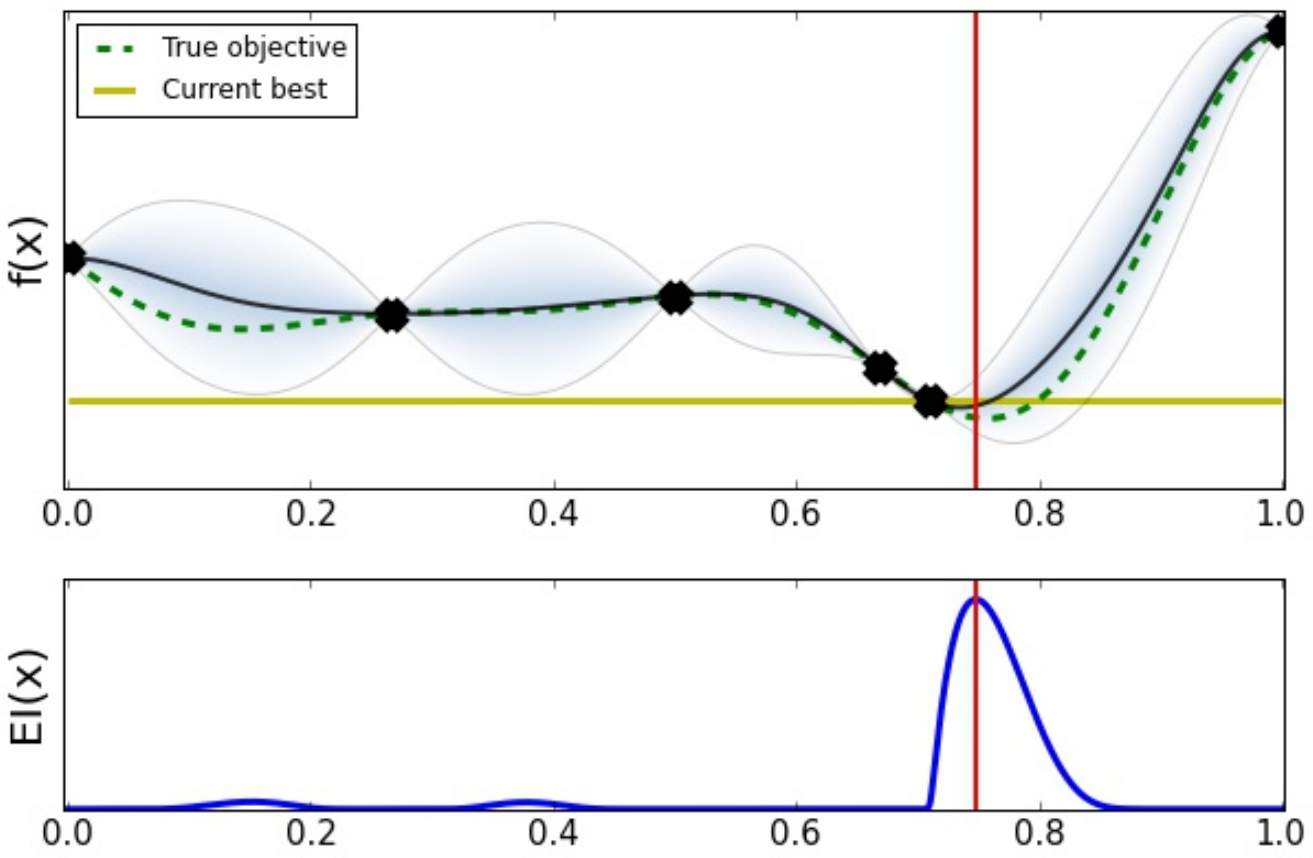


Illustration of BO

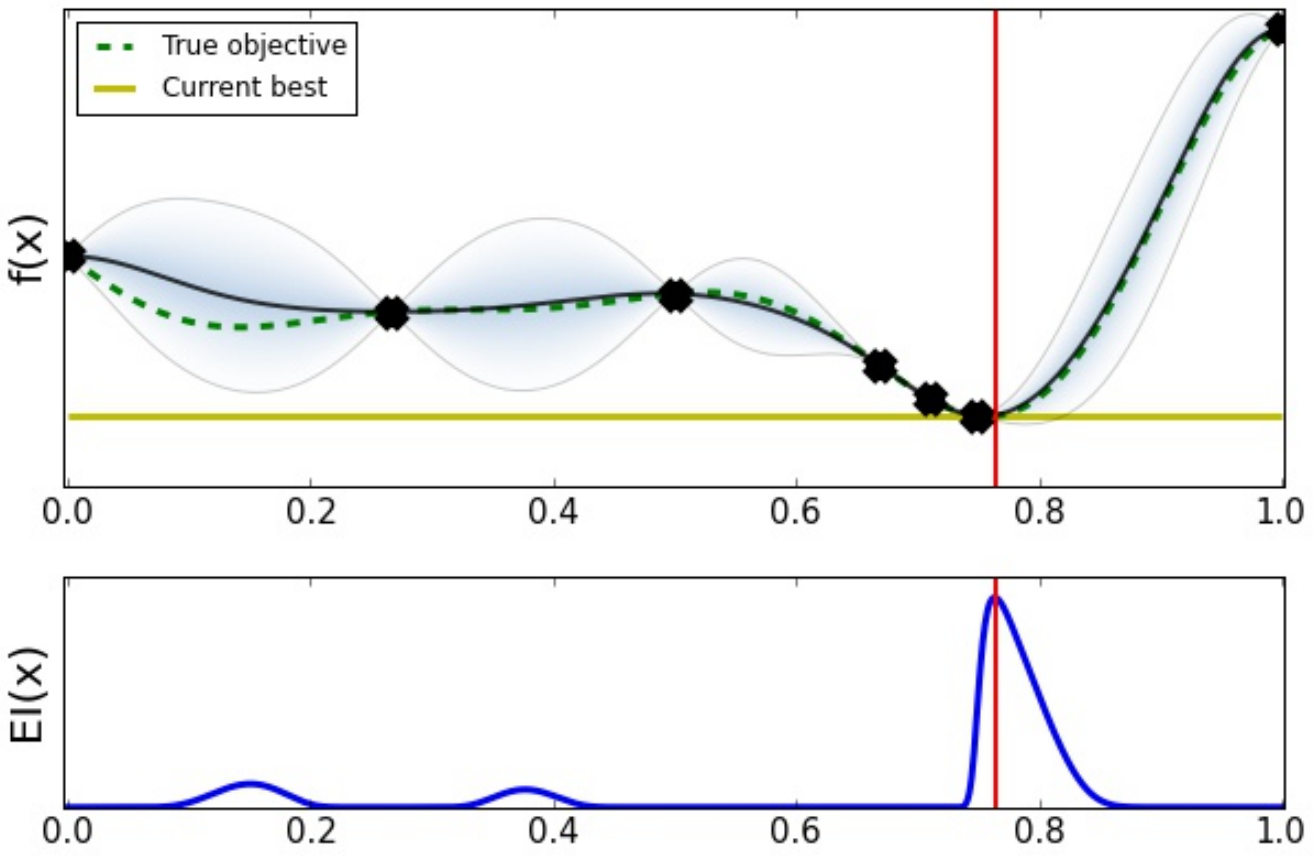
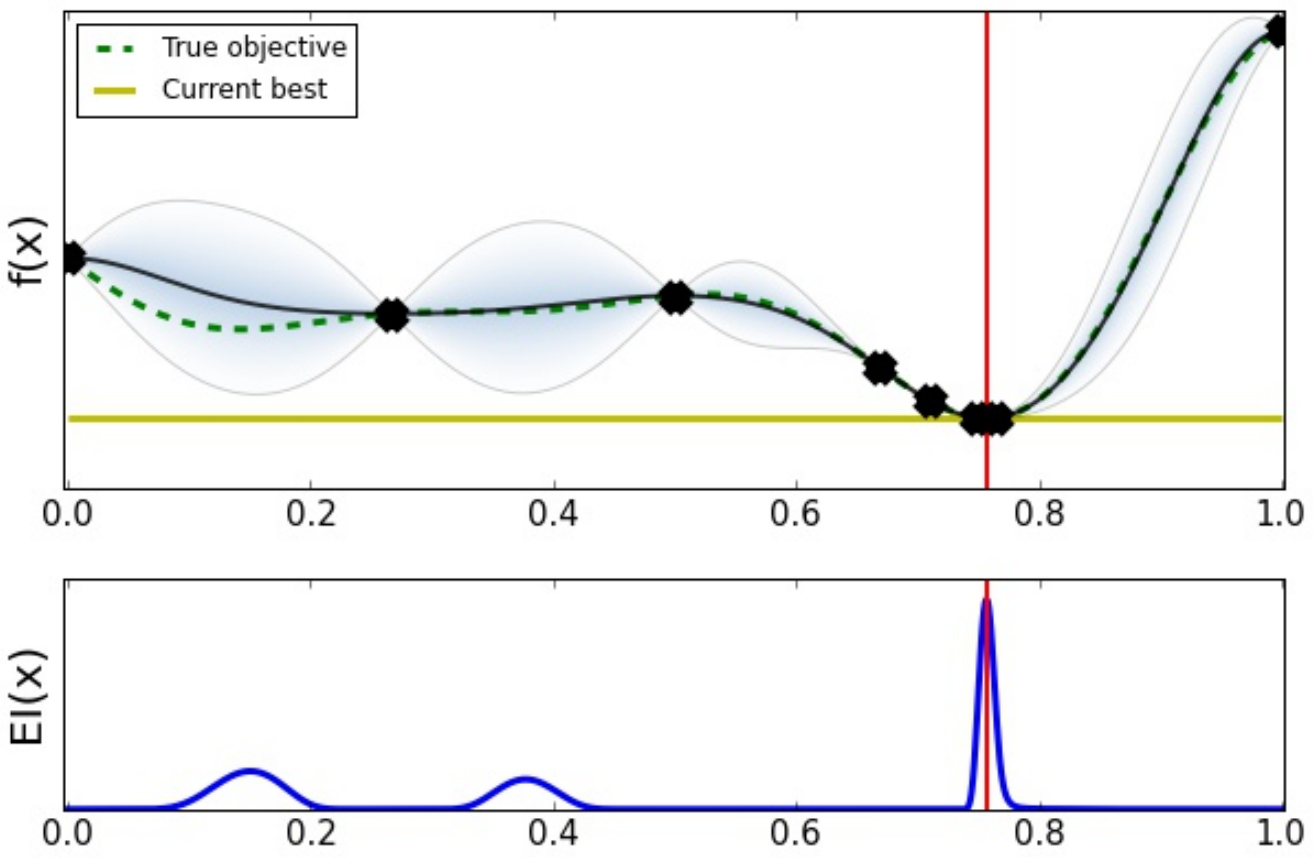


Illustration of BO



Bayesian Optimization

As a 'mapping' between two problems

BO is an strategy to transform the problem

$$x_M = \arg\min_{x \in \mathcal{X}} f(x)$$

solvable!

into a series of problems:

$$x_{n+1} = \arg\max_{x \in \mathcal{X}} \alpha(x; \mathcal{D}_n, \mathcal{M}_n)$$

solvable!

where now:

- $\alpha(x)$ is inexpensive to evaluate.
- The gradients of $\alpha(x)$ are typically available.
- Still need to find x_{n+1}.

BO vs other methods

[Osborne et al, 2009]

Bayesian optimization works better in practice!

				GPGO 1-Step		GPGO 2-Step
	EGO	RBF	DIRECT	Non-Periodic	Periodic	Non-Periodic
Br	0.943	0.960	0.958	**0.980**	—	—
C6	0.962	0.962	0.940	0.890	—	**0.967**
G–P	0.783	0.815	**0.989**	0.804	—	**0.989**
H3	0.970	0.867	0.868	**0.980**	—	—
H6	0.837	0.701	0.689	**0.999**	—	—
Sh5	0.218	0.092	0.090	**0.485**	—	—
Sh7	0.159	0.102	0.099	**0.650**	—	—
Sh10	0.135	0.100	0.100	**0.591**	—	—
GK2	0.571	0.567	0.538	**0.643**	—	—
GK3	0.519	0.207	0.368	**0.532**	—	—
Shu	**0.492**	0.383	0.396	0.437	0.348	0.348
G2	0.979	**1.000**	0.981	**1.000**	**1.000**	—
G5	**1.000**	0.998	0.908	0.925	0.957	—
A2	0.347	0.703	0.675	0.606	0.612	**0.781**
A5	0.192	**0.381**	0.295	0.089	0.161	—
R	0.652	0.647	0.776	0.675	**0.933**	—
mean	0.610	0.593	0.604	**0.705**	—	—

Recap

- Bayesian optimization is a way of encoding our beliefs about a property of a function (the minimum)
- Two key elements: the model and the acquisition function.
- Many choices in both cases, especially in terms of the acquisition function used.
- The key is to find a good balance between exploration and exploitation.

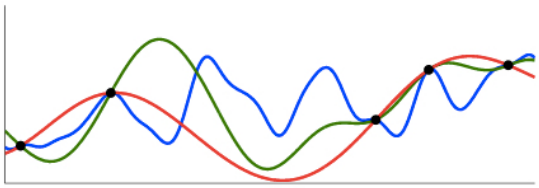
Main issues

- What to do with the hyper-parameters of the model?
- How to select points to initialize the model?
- How to optimize the acquisition function?

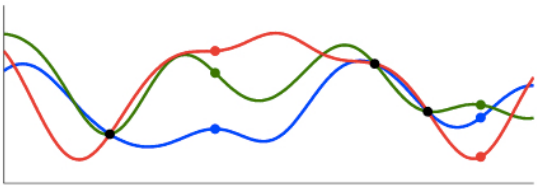
BO independent of the parameters of the GP.

[Snoek et al. 2012]

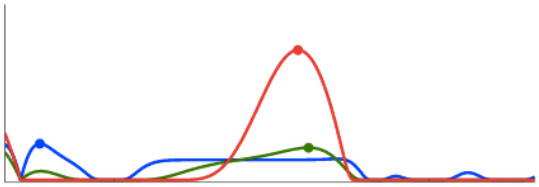
Integrate out across parameter values or location outputs.



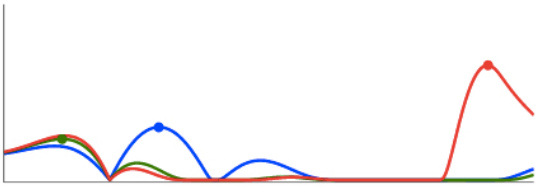
(a) Posterior samples under varying hyperparameters



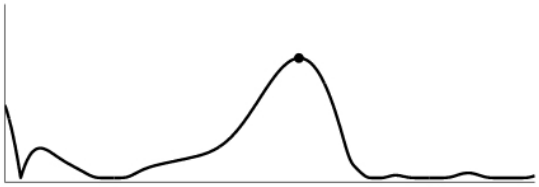
(a) Posterior samples after three data



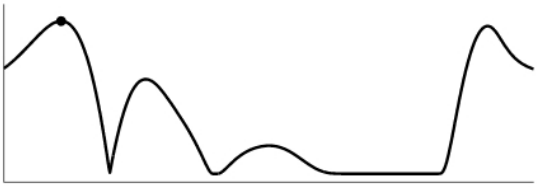
(b) Expected improvement under varying hyperparameters



(b) Expected improvement under three fantasies



(c) Integrated expected improvement



(c) Expected improvement across fantasies

How to initialise the model?

- One point in the centre of the domain.
- Uniformly selected random locations.
- Latin design.
- Halton sequences.
- Determinantal point processes.

The idea is always to start at some locations trying to minimise the initial model uncertainty.

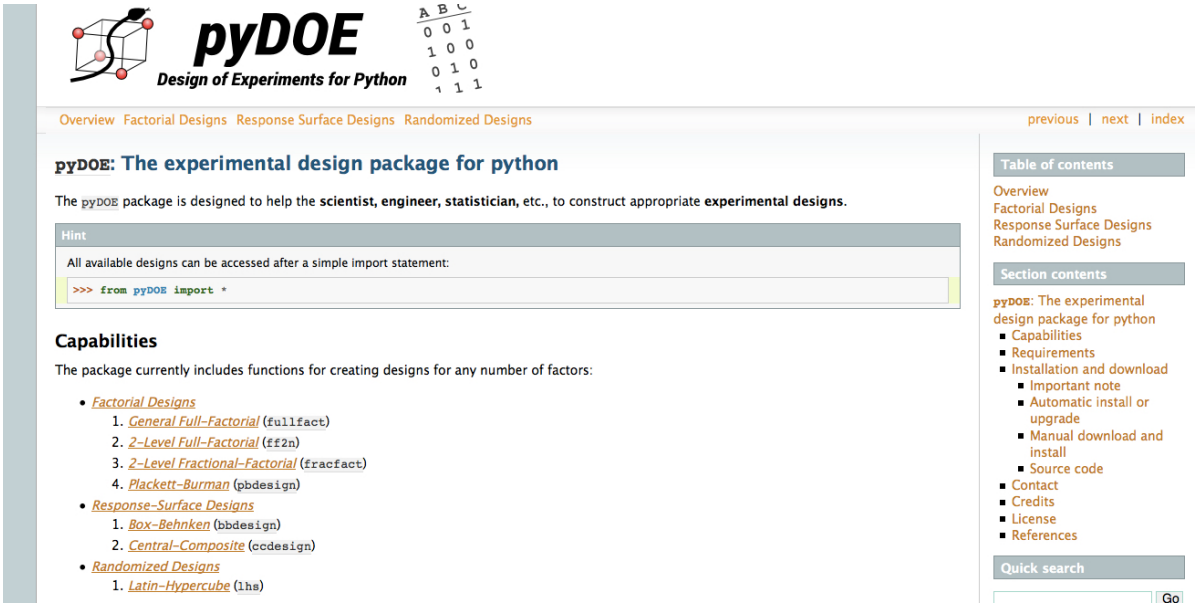
Latin design

$n \times n$ array filled with n different symbols, each occurring exactly once in each row and exactly once in each column.

A	B	F	C	E	D
B	C	A	D	F	E
C	D	B	E	A	F
D	E	C	F	B	A
E	F	D	A	C	B
F	A	E	B	D	C

pyDOE

Python framework for standard experimental design



pyDOE
Design of Experiments for Python

Overview Factorial Designs Response Surface Designs Randomized Designs

previous | next | index

pyDOE: The experimental design package for python

The pyDOE package is designed to help the **scientist, engineer, statistician,** etc., to construct appropriate **experimental designs**.

Hint

All available designs can be accessed after a simple import statement:

```
>>> from pyDOE import *
```

Capabilities

The package currently includes functions for creating designs for any number of factors:

- *Factorial Designs*
  1. *General Full-Factorial* (`fullfact`)
  2. *2-Level Full-Factorial* (`ff2n`)
  3. *2-Level Fractional-Factorial* (`fracfact`)
  4. *Plackett-Burman* (`pbdesign`)
- *Response-Surface Designs*
  1. *Box-Behnken* (`bbdesign`)
  2. *Central-Composite* (`ccdesign`)
- *Randomized Designs*
  1. *Latin-Hypercube* (`lhs`)

Table of contents

Overview
Factorial Designs
Response Surface Designs
Randomized Designs

Section contents

pyDOE: The experimental design package for python
- Capabilities
- Requirements
- Installation and download
  - Important note
  - Automatic install or upgrade
  - Manual download and install
  - Source code
- Contact
- Credits
- License
- References

Quick search

Go

Latin design

Window honors Ronald Fisher. Fisher's student, A. W. F. Edwards, designed this window for Caius College, Cambridge.

Halton sequences

[Halton, 1964]

- Used to generate points in $(0, 1) \times (0, 1)$
- Sequence that is constructed according to a deterministic method that uses a prime number as its base.

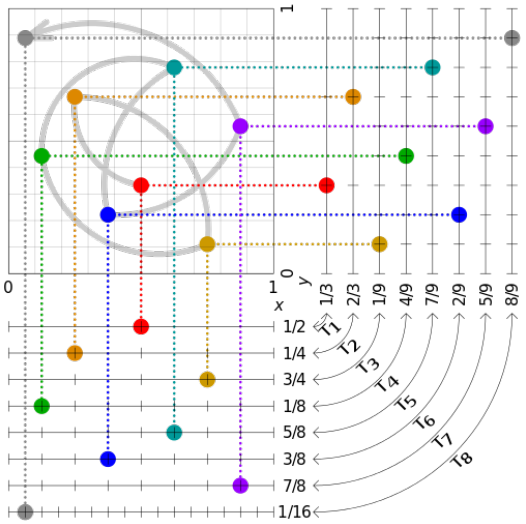
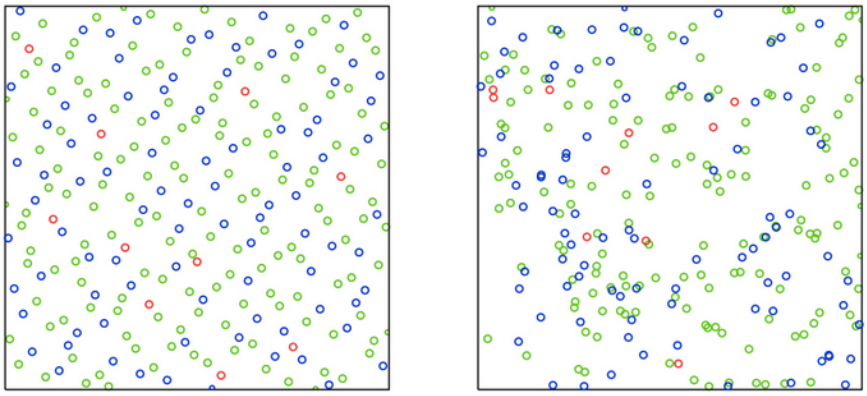


Figure source: Wikipedia

Halton sequences

[Halton, 1964]

Better coverage than random.



Halton *Random*

Figure source: Wikipedia

Determinantal point processes

Kulesza and Taskar, [2012]

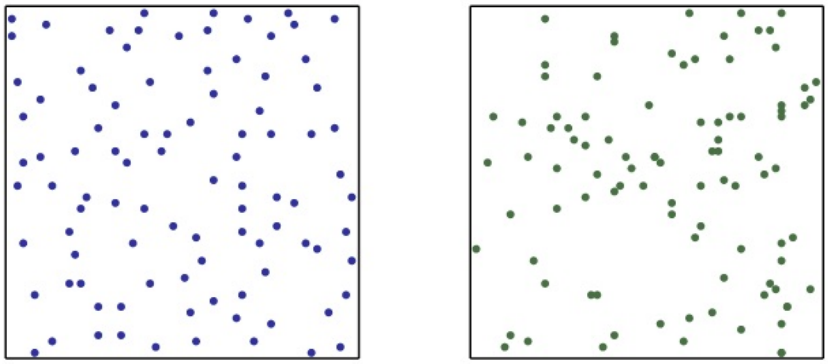
We say that X is a 'determinantal point process' on Λ with kernel K if it is a simple point process on Λ with a joint intensity or 'correlation function' given by

$$\rho_n(x_1, \ldots, x_n) = det(K(x_i, x_j)_{1 \leq i,j \leq n})$$

- Probability measures over subsets.
- Possible to characterise the samples in terms of quality and diversity.

Determinantal point processes

Kulesza and Taskar, [2012]



Key idea:

$$\begin{aligned}\mathcal{P}(i,j \in \boldsymbol{Y}) &= \begin{vmatrix} K_{ii} & K_{ij} \\ K_{ji} & K_{jj} \end{vmatrix} \\ &= K_{ii}K_{jj} - K_{ij}K_{ji} \\ &= \mathcal{P}(i \in \boldsymbol{Y})\mathcal{P}(j \in \boldsymbol{Y}) - K_{ij}^2 \,.\end{aligned}$$

Determinantal point processes

Kulesza and Taskar, [2012]

Methods to optimise the acquisition function

This may not be easy.

- Gradient descent methods: Conjugate gradient, BFGS, etc.
- Lipschitz based heuristics: DIRECT.
- Evolutionary algorithms: CMA.

Some of these methods can also be used to directly optimize f

Gradient descent

[Avriel,2013], but many others

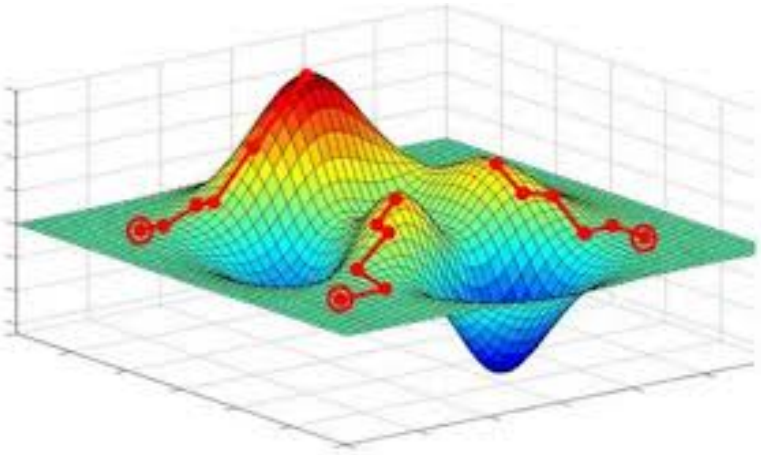
Algorithm 2: Gradient Descent

input : $f : \mathbb{R}^n \to \mathbb{R}$ a differentiable function
$\mathbf{x}^{(0)}$ an initial solution
output: $\mathbf{x}^\star$, a local minimum of the cost function f.

begin
  $k \leftarrow 0$;
  while STOP-CRIT **and** $(k < k_{max})$ **do**
    $\mathbf{x}^{(k+1)} \leftarrow \mathbf{x}^{(k)} - \alpha^{(k)} \nabla f(\mathbf{x})$;
    with $\alpha^{(k)} = \arg\min_{\alpha \in \mathbb{R}_+} f\left(\mathbf{x}^{(k)} - \alpha \nabla f(\mathbf{x})\right)$;
    $k \leftarrow k + 1$;
  return $\mathbf{x}^{(k)}$
end

We need to know the gradients. This is the case for most acquisitions but not for all of them (PES for instance).

Gradient descent



May fall in local minima if the function is multimodal: multiple initializations.

‘DIviding RECTangles’, DIRECT

[Perttunen at al. 1993]

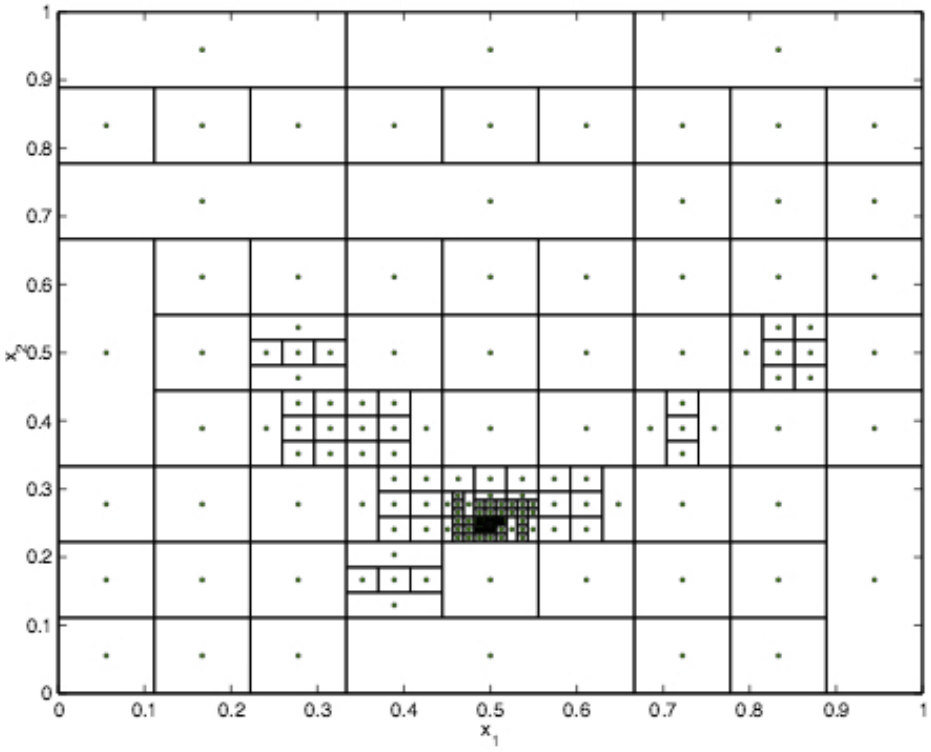
Algorithm `DIRECT`('myfcn',bounds,opts)

Normalize the domain to be the unit hyper-cube with center c_1
Find $f(c_1)$, $f_{min} = f(c_1)$, $i = 0$, $m = 1$
Evaluate $f(c_1 \pm \delta e_i$, $1 \leq i \leq n$, and divide hyper-cube
while $i \leq maxits$ and $m \leq maxevals$ **do**
    Identify the set S of all pot. optimal rectangles/cubes
    for all $j \in S$
        Identify the longest side(s) of rectangle j
        Evaluate myfcn at centers of new rectangles, and divide j into smaller rectangles
        Update f_{min}, $xatmin$, and m
    end for
    $i = i + 1$
end while

Minimal hypothesis about the acquisition

‘DIviding RECTangles’, DIRECT

[Perttunen at al. 1993]

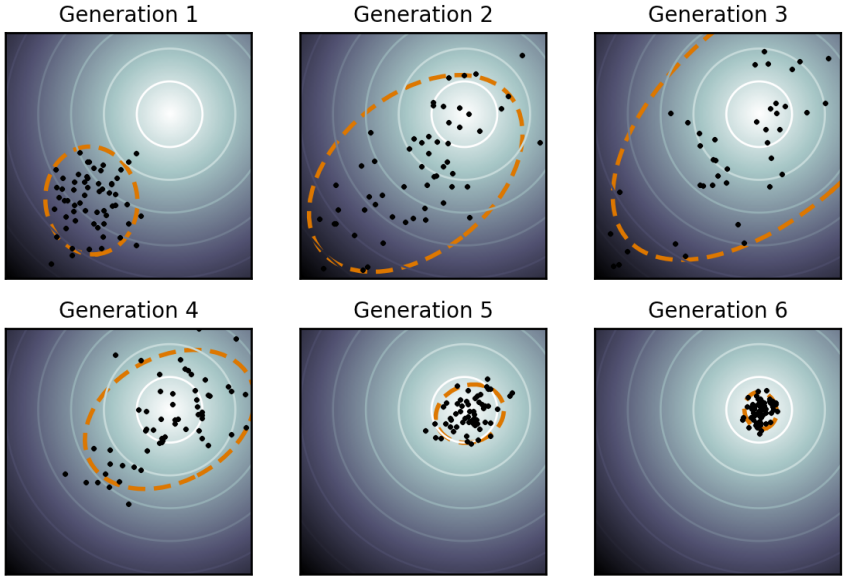


Finds good solution in general and doesn’t need gradient. Not generalizable to non-squared domains.

Covariance Matrix Adaptation, CMA

[Hansen and Ostermeier, 2001].

- Sample for a Gaussian with some mean μ and covariance matrix Σ.
- Select the best points and use them to update μ and Σ.
- Sample form the new Gaussian.



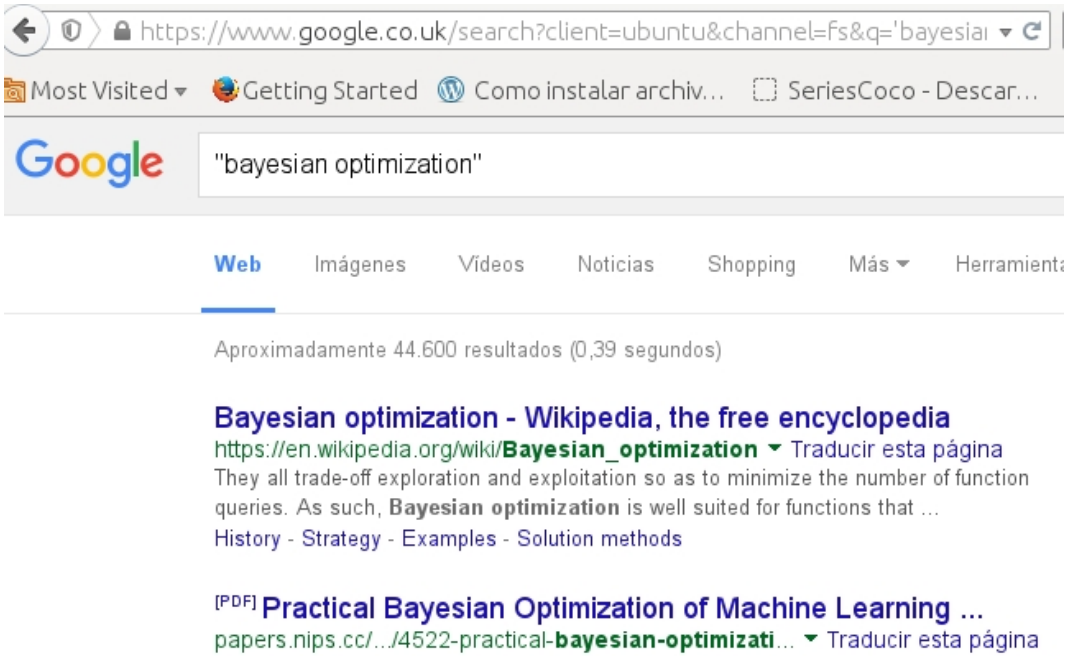
Took a while to start using these ideas in ML

Although in the stats community have been there for a while

- BO depends on its own parameters.
- Lack of software to apply these methods as a black optimization boxes.
- Reduced scalability in dimensions and number of evaluations (this is still a problem).

Practical Bayesian Optimization of Machine Learning Algorithms. Snoek, Larochelle and Adams. NIPS 2012 (Spearmint)

Increasing popular field



- Hot topic in Machine Learning.
- The BO workshop at NIPS is well stablished and it is a mini-conference itself.

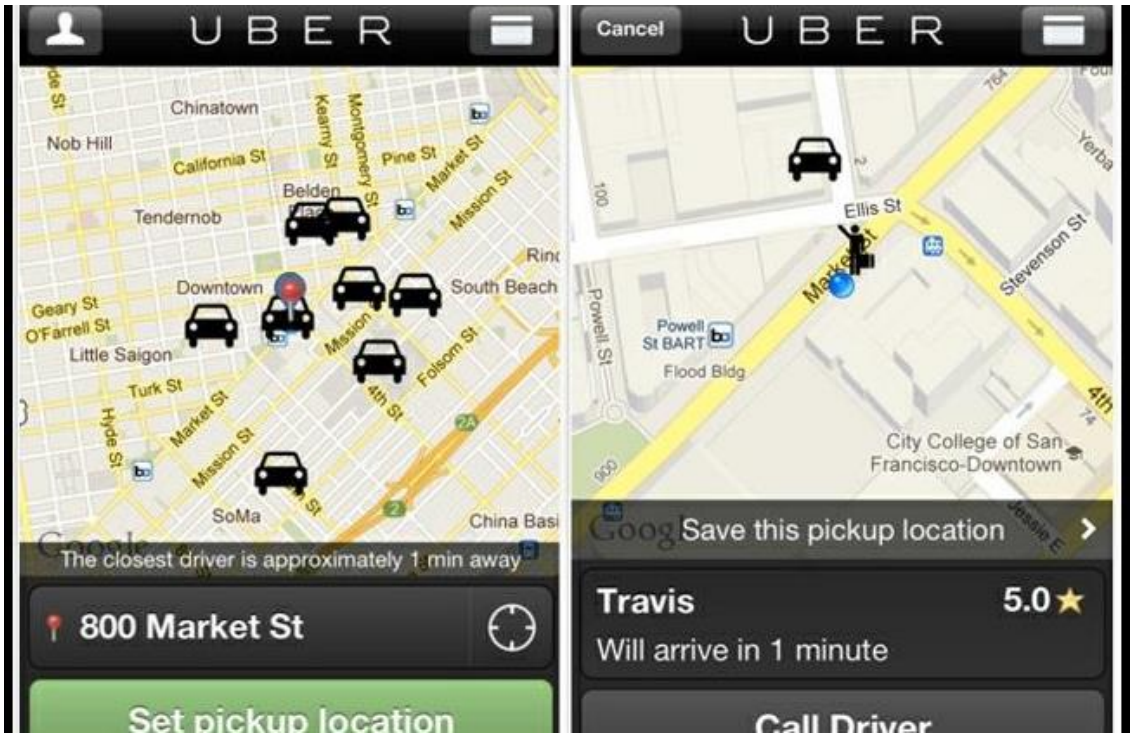
Bayesian optimization now

It has become increasingly popular since it allows to configure algorithms without human intervention.

BO takes to human out of the loop!

BO in industry: Twitter

BO in industry: Uber



Questions?